영화 리뷰 감성 분석

: 영화 추천 및 새로운 영화 도입에 대한 판단 기준 제언

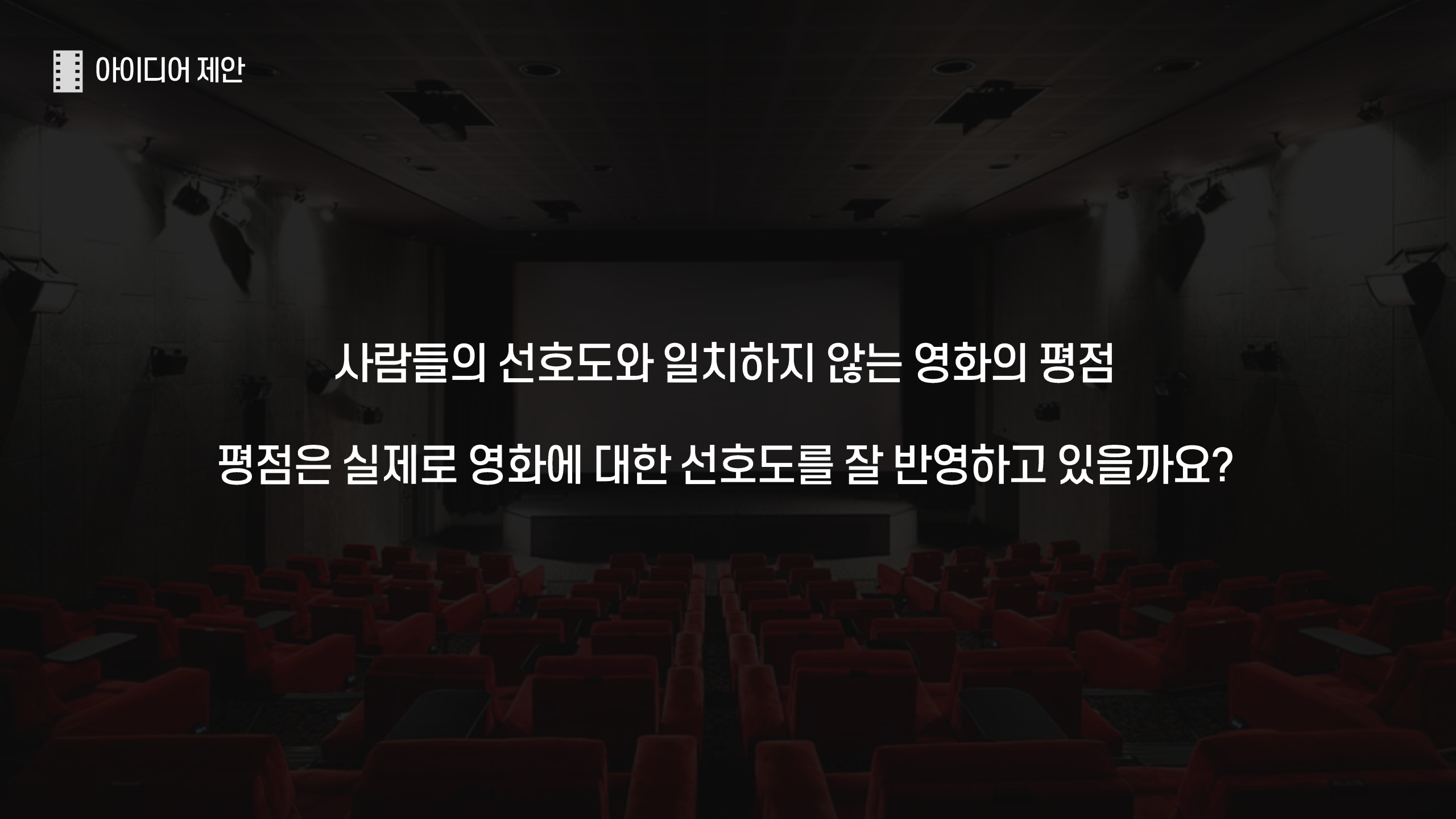
오코사 (오늘도 코딩하는 사람들)
: 김나래, 박서희, 박채린, 박호준

Index

역할분담

김나래	팀장, 데이터 수집, 모형 분석 및 설명
박서희	데이터 수집, 데이터 전처리, 데이터 시각화
박채린	데이터 수집, 데이터 탐색 및 통합, PPT
박호준	데이터 수집, 데이터 전처리, 모형 분석 및 설명

아이디어 배경



아이디어 제안

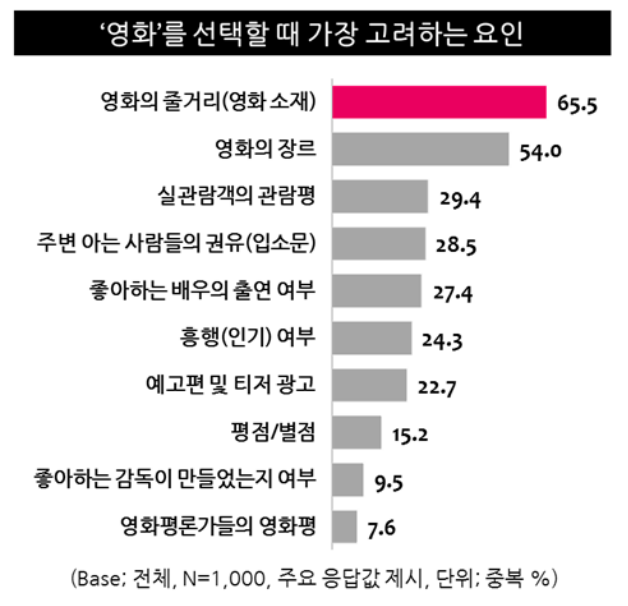
사람들의 선호도와 일치하지 않는 영화의 평점

평점은 실제로 영화에 대한 선호도를 잘 반영하고 있을까요?

아이디어 배경

영화 선택 기준으로 고려되는 평점(★★★★★), 하지만 실제 호감도와 평점은 일치하지 않을 수도 있습니다.

영화 선택 고려 요인 분석



줄거리 > 장르 > 관람평 > 평점 순

평점과 영화 선호도 불일치

무분별한 평점 테러 논란, 평점 게시판의 현주소

평점 테러는 영화 관람 여부와 관계없이 **고의로 낮은 평점을 부과해 평점을 낮추는 행위**를 말한다.…
영화 '걸캅스'와 '캡틴 마블'은 젠더 간 **갈등으로 평점 테러를 겪었지만, 손익분기점을 넘기는 등 흥행에 성공**했다.
지난 5월 개봉한 영화 '걸캅스'는 불법 촬영 범죄를 척결하는 두 여성 경찰을 주연으로 한다는 점 때문에 일부 네티즌으로부터 평점 테러를 당했다.
지난 3월 개봉한 영화 '캡틴 마블'은 주연 배우 브리 라슨의 "위대한 페미니스트의 영화가 될 것"이라는 발언이 논란이 돼 평점 테러의 대상이 됐다.

출처: 숭대시보

평점에 대한 **신뢰성**에 문제가 제기됨

아이디어 배경

반면 리뷰는 관람자가 느끼는 감정을 담고 있는 데이터이기에 영화를 선택하는 사람들에게 도움을 줄 수 있습니다

감정을 나타내는 리뷰들

영화에 대해 **긍정적**인 리뷰

국뽕이라도 상관없다! 볼 때마다 격한 감동
넘재미 있고 울컥 울컥 하네요

영화에 대해 **부정적**인 리뷰

사람들이 영화에 대한 평점이 아닌 이순신에 대한 평점을 매기는구나 국뽕에 눈 돌아가는 것 좀 자제했으면

감성분석의 실 사례

'상대의 속마음이 궁금하다'…
글 속에 숨은 호감도 알아내는 AI

LPG 팀은 코로나19 확산 이후 큰 성장세를 기록하고 있는 국내외 데이팅 앱 시장에 주목해 결혼 플랫폼 기업인 ㈜여보야의 … 사용자들 간의 채팅 후 '마음의 소리' **글을 통해 속마음을 분석함으로써 상대방의 호감도**를 **알려줘** 매칭 확률을 높이겠다는 구상이다.

LPG 팀은 **텍스트 데이터 분석**을 **통해** 서로의 **호감을 확인**하고 상대방과 만남을 지속할지에 대한 속마음을 간접적으로 전달하는 기능을 차별화된 강점으로 내세웠다.

출처: AI 타임스

프로젝트 목표

반면 리뷰는 관람자가 느끼는 감정을 담고 있는 데이터이기에
영화를 선택하는 사람들에게 도움을 줄 수 있습니다

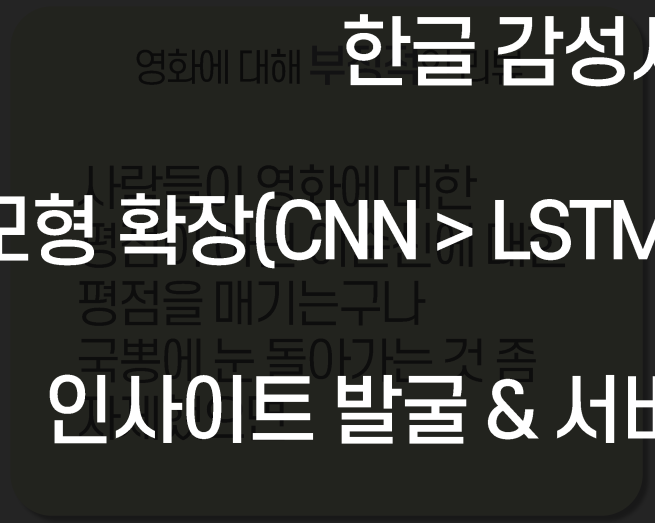
•

여러 방법의 토크나이징 방법 도입

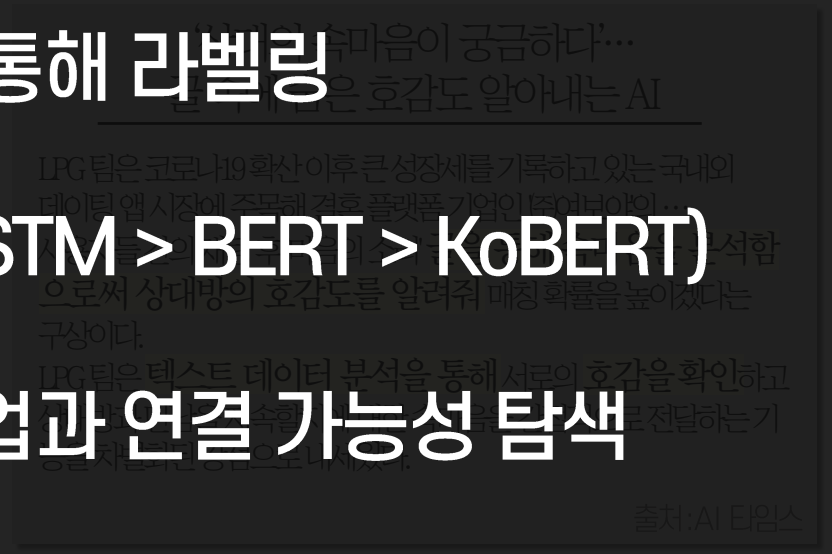
한글 감성사전을 통해 라벨링

모형 확장(CNN > LSTM > BiLSTM > BERT > KoBERT)

인사이트 발굴 & 서비스 산업과 연결 가능성 탐색



영화 리뷰를 통해
관람자가 느끼는 감정을
파악할 수 있음



비정형 텍스트에 대해
감성분석을 함으로써
호감도를 파악할 수 있음

프로젝트 진행 flow

데이터 수집

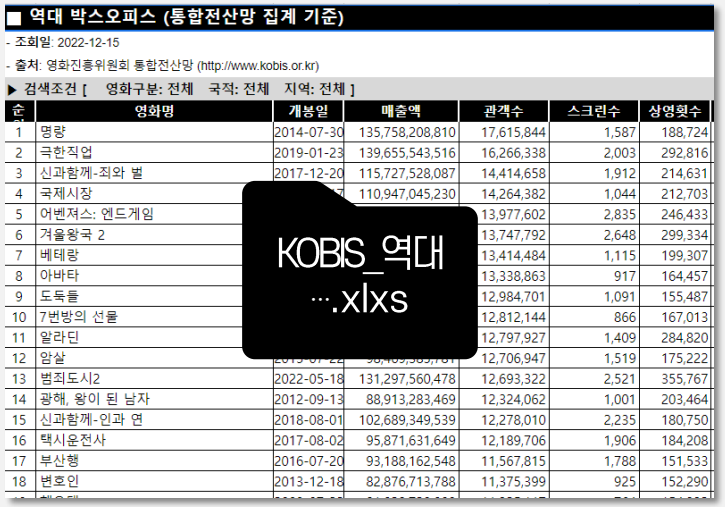
데이터 수집

KOBIS에서 제공하는 역대 박스오피스를 참고하여 TOP 100에 해당하는 영화를 선정하였습니다

영화관입장권 통합전산망 KOBIS

전국영화관 입장권 발권정보를
실시간으로 집계 / 처리하는 시스템
KOBIS(서비스 플랫폼)

역대 박스오피스 TOP 100 영화



■ 역대 박스오피스 (통합전산망 집계 기준)

- 조회일: 2022-12-15
- 출처: 영화진흥위원회 통합전산망 (http://www.kobis.or.kr)

▶ 검색조건 [영화구분: 전체 국적: 전체 지역: 전체]

순	영화명	개봉일	매출액	관객수	스크린수	상영횟수
1	명량	2014-07-30	135,758,208,810	17,615,844	1,587	188,724
2	극한직업	2019-01-23	139,655,543,516	16,266,338	2,003	292,816
3	신과함께-죄와 벌	2017-12-20	115,727,528,087	14,414,658	1,912	214,631
4	국제시장		110,947,045,230	14,264,382	1,044	212,703
5	어벤져스: 엔드게임			13,977,602	2,835	246,433
6	겨울왕국 2			13,747,792	2,648	299,334
7	베테랑			13,414,484	1,115	199,307
8	아바타			13,338,863	917	164,457
9	도둑들			12,984,701	1,091	155,487
10	7번방의 선물			12,812,144	866	167,013
11	알라딘			12,797,927	1,409	284,820
12	암살			12,706,947	1,519	175,222
13	범죄도시2	2022-05-18	131,297,560,478	12,693,322	2,521	355,767
14	광해, 왕이 된 남자	2012-09-13	88,913,283,469	12,324,062	1,001	203,464
15	신과함께-인과 연	2018-08-01	102,689,349,539	12,278,010	2,235	180,750
16	택시운전사	2017-08-02	95,871,631,649	12,189,706	1,906	184,208
17	부산행	2016-07-20	93,188,162,548	11,567,815	1,788	151,533
18	변호인	2013-12-18	82,876,713,788	11,375,399	925	152,290

KOBIS_역대
….xlxs

역대 박스오피스 TOP 100 영화
영화명과 개봉일 수집하여 사용

네이버, 왓챠피디아, 다음에서 selenium을 활용하여 TOP100 영화의 리뷰와 평점 정보를 크롤링했습니다

영화 리뷰 서비스 제공 사이트

가장 많은 리뷰 데이터를 가지고 있는
왓챠, 네이버, 다음에서
영화명, 리뷰 , 평점에 대한
데이터 크롤링 진행

크롤링 코드 일부 및 csv 파일 생성

```
# 평점 탭 클릭
sel = 'div.tabmenu_wrap > ul > li > a > span'
driver.find_elements(By.CSS_SELECTOR, sel)[3].click()
time.sleep(2)

# 정
sel2
drive
time.

# 평
review_more_click(more_num)

# 한 페이지의 리뷰 전체 불러오기
sel3 = '#alex-area > div > div > div > div.cmt_box > ul.list_comment > li'
reviews = driver.find_elements(By.CSS_SELECTOR, sel3)
```

review_data_naver.csv

review_data_daum.csv

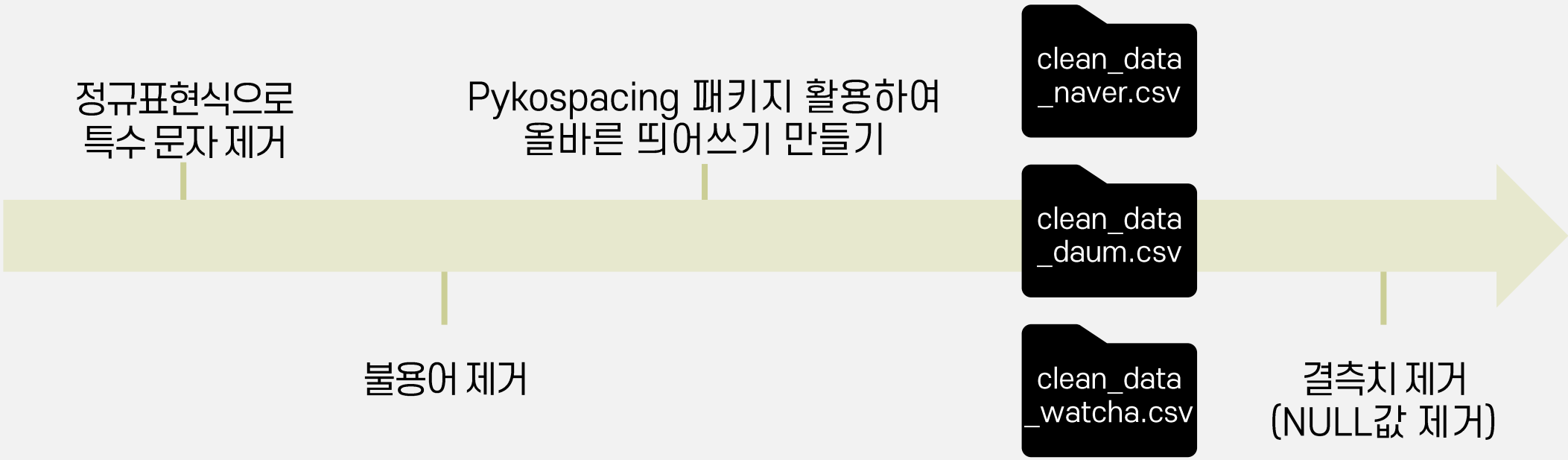
review_data_watcha.csv

Selenium 크롤링을 사용하여
각 사이트별 데이터를 csv 파일로 저장

데이터 전처리

리뷰 데이터 탐색에 앞서
데이터 전처리를 통해 정제 데이터를 생성하였습니다

데이터 정제 과정 도식화

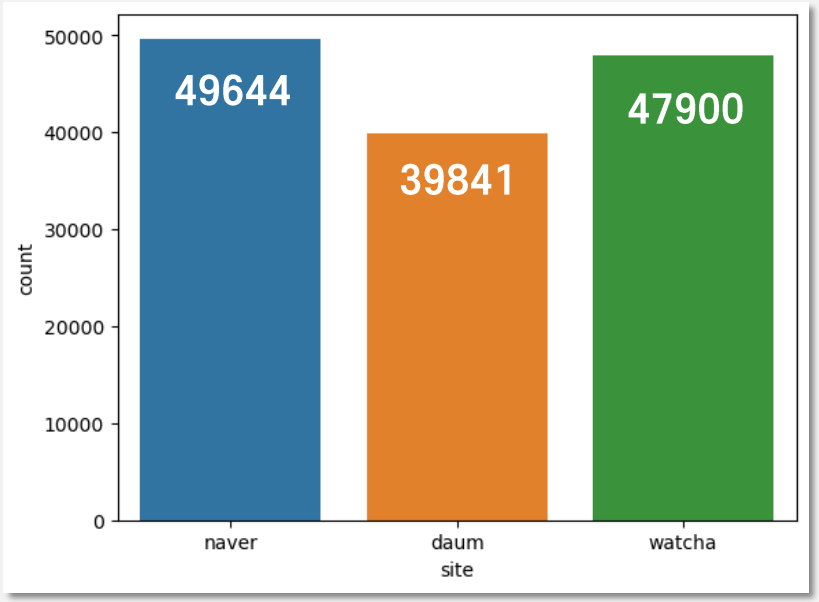


데이터 탐색 및 병합

데이터 탐색 및 병합

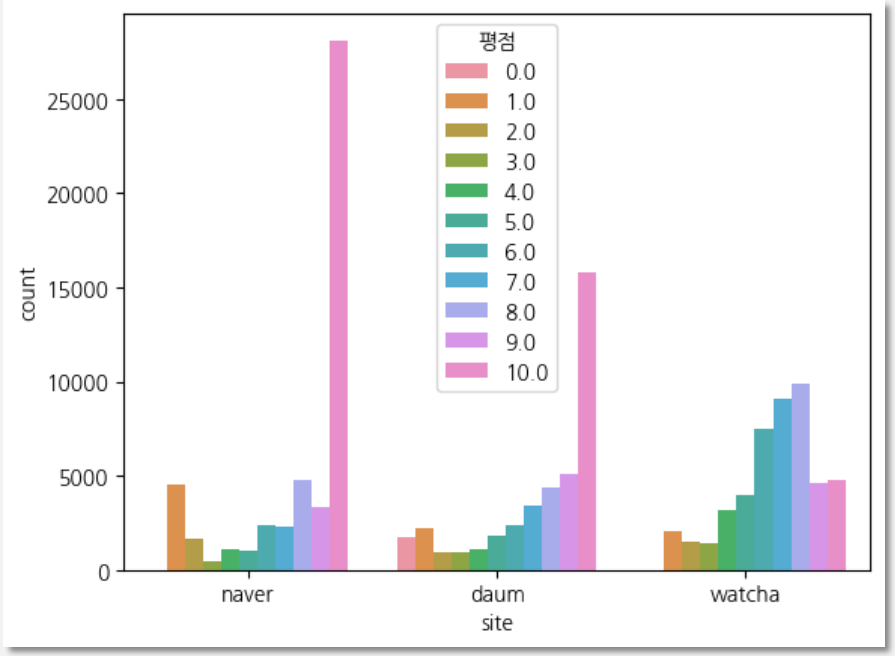
수집한 리뷰에 대한 인사이트를 얻기 위해
정제된 데이터를 활용하여 탐색 (EDA) 을 진행하였습니다

각 사이트별 리뷰 수 count



네이버 다음 왓챠

각 사이트별 평점 분포

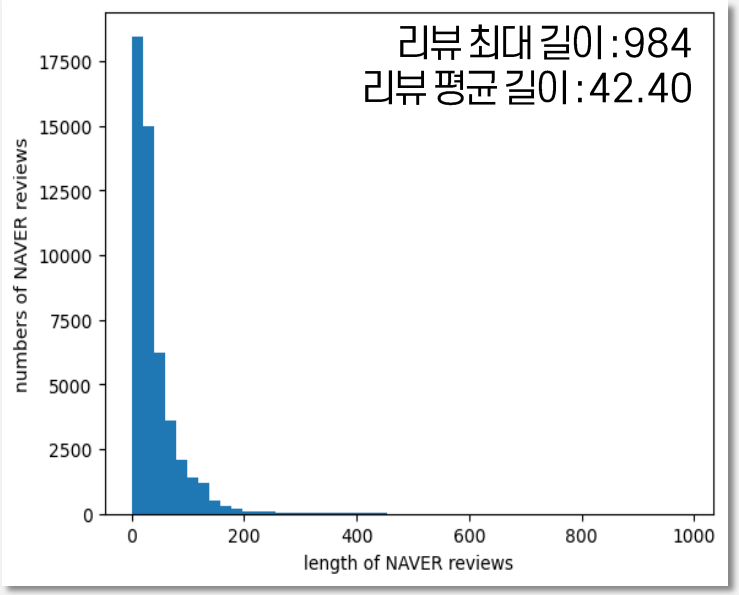


네이버 다음 왓챠

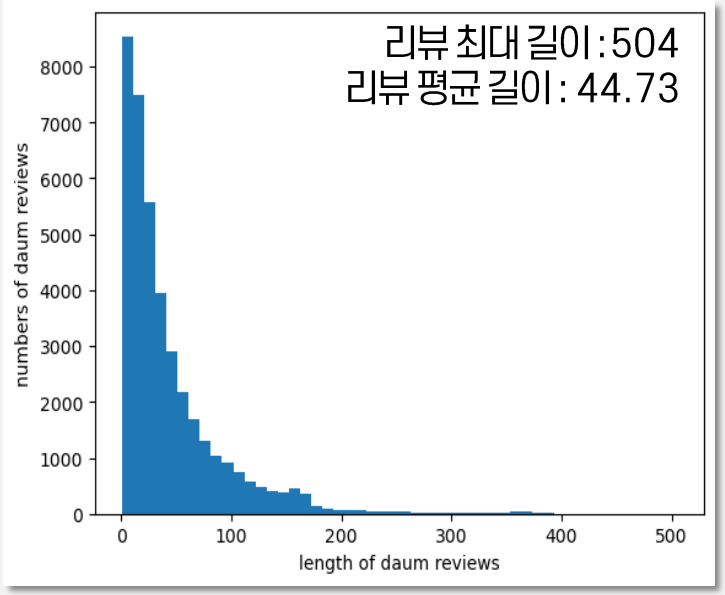
데이터 탐색 및 병합

수집한 리뷰에 대한 인사이트를 얻기 위해 정제된 데이터를 활용하여 탐색 (EDA) 을 진행하였습니다

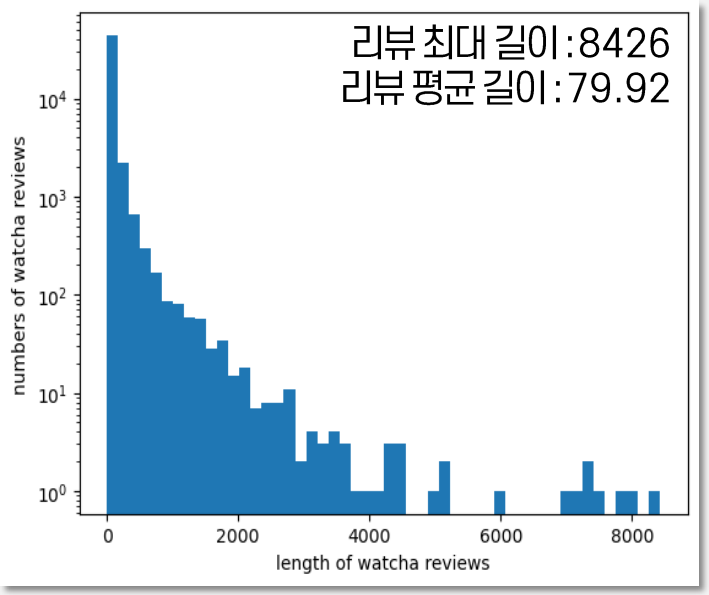
네이버 리뷰 길이 시각화



다음 리뷰 길이 시각화

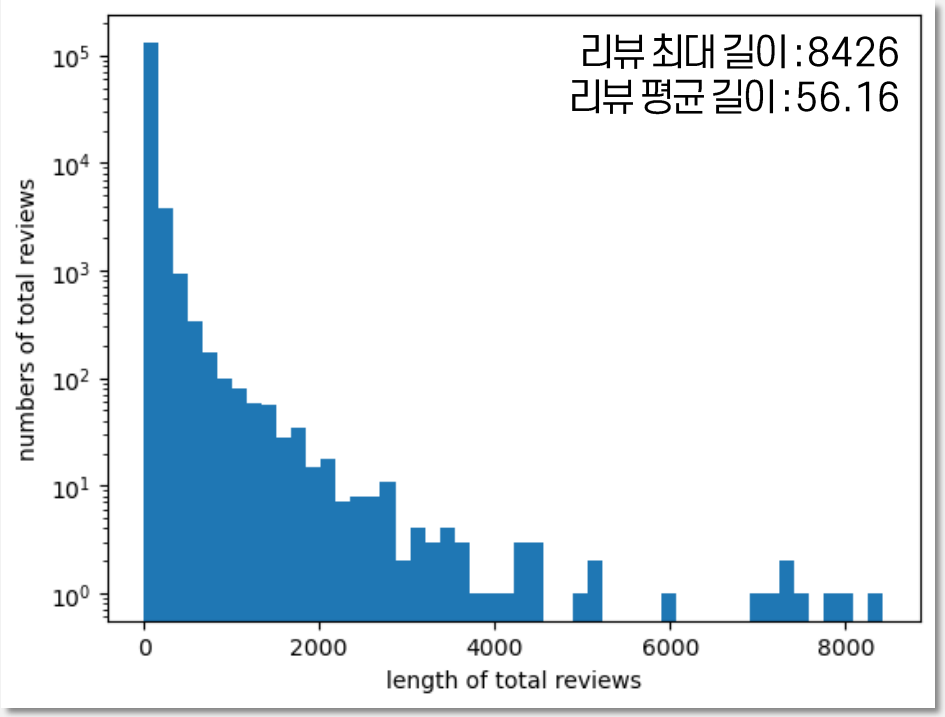


왓챠 리뷰 길이 시각화

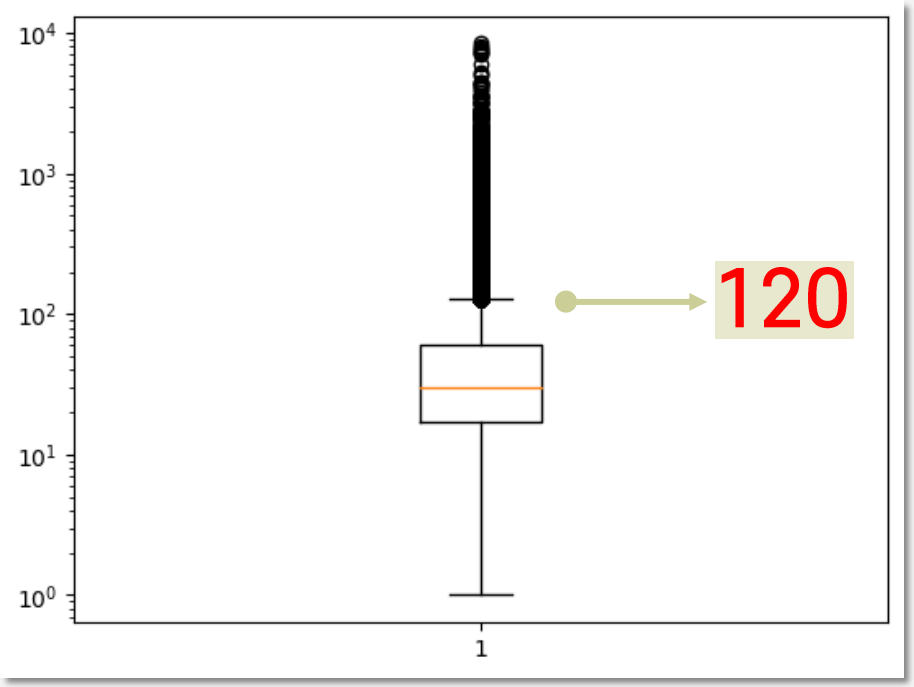


데이터 병합 후 전체 리뷰의 길이 분포에 대한 시각화를 통해 리뷰 최대 길이를 제한합니다

전체 리뷰 길이 히스토그램

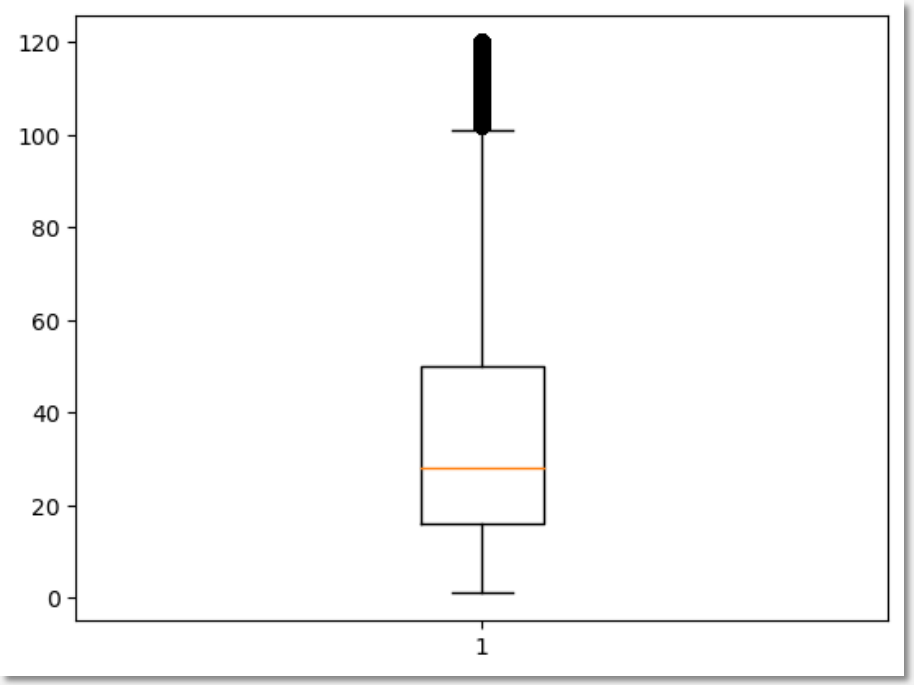


전체 리뷰의 길이 박스플롯



최대 리뷰 길이를 120으로 필터링한 최종 데이터 셋에 대해 간단한 탐색을 진행합니다.

전체 리뷰의 길이 박스플롯



명사 워드 클라우드

1위 영화,
2위 연기,
3위 것

최대 리뷰 길이를 120으로 필터링한
최종 데이터 셋에 대해 간단한 탐색을 진행합니다.

형태소 워드 클라우드

1위 이,
2위 영화,
3위 의

어절 워드 클라우드

1위 영화,
2위 연기,
3위 진짜

형태소 분석

보편적으로 사용되는 konlpy의 형태소 분석기의 한계를 발견.
신조어 및 복합명사 등의 미등록 단어들을 사전에 추가하고자 함.

Konlpy 형태소 분석기

▶ Okt, Kkma, Hannum, Mecab...

▶ **Okt**

장점> 상대적으로 빠르다
다양하게 사전추가가 가능하다

단점> 신조어 및 복합명사에 취약하다

▶ Kkma

장점> 품사를 세세하게 나눌 수 있다.

단점> 상대적으로 느리다
너무 세세해서 과하게 점수가 적용된다

< Example >

(명사 추가)
한국영화(한국영 / 화), 씨지(씨/지)

(이름 추가)
이순신, 마블리, 톰형, 민식이형.....

(신조어 추가)
꾸르잼, 머시쩡, 파괴왕, 뽀짝, 국뽕 ....

사전 추가를 위해 Soynlp를 사용하여 Okt의 단어사전과 비교할 명사 데이터를 준비합니다

Soynlp를 활용한 명사추출

	nouns	frequency	score
7207	영화	23963.0	0.839022
7324	너무	7225.0	0.855072
7262	생각	3758.0	0.977139
7335	재미	3539.0	0.944164
10553	정말	3388.0	1.000000
...	...	...	...
17755	어벤져스가입	1.0	1.000000
17756	인간성진부	1.0	1.000000
17757	한국도박영화	1.0	1.000000
17758	재구성도둑들	1.0	1.000000
25886	조화사회	1.0	1.000000

25887 rows × 3 columns

리뷰 데이터셋에서 soynlp를 활용하여
명사를 추출한 결과를
출현 빈도순으로 나열한 결과

정규표현식으로
숫자와 모음 제거

Okt를 사용하여
품사태깅

Review에서 soynlp가 추출한 명사들을 정리하는 과정

공백과 한글자인
명사 제거

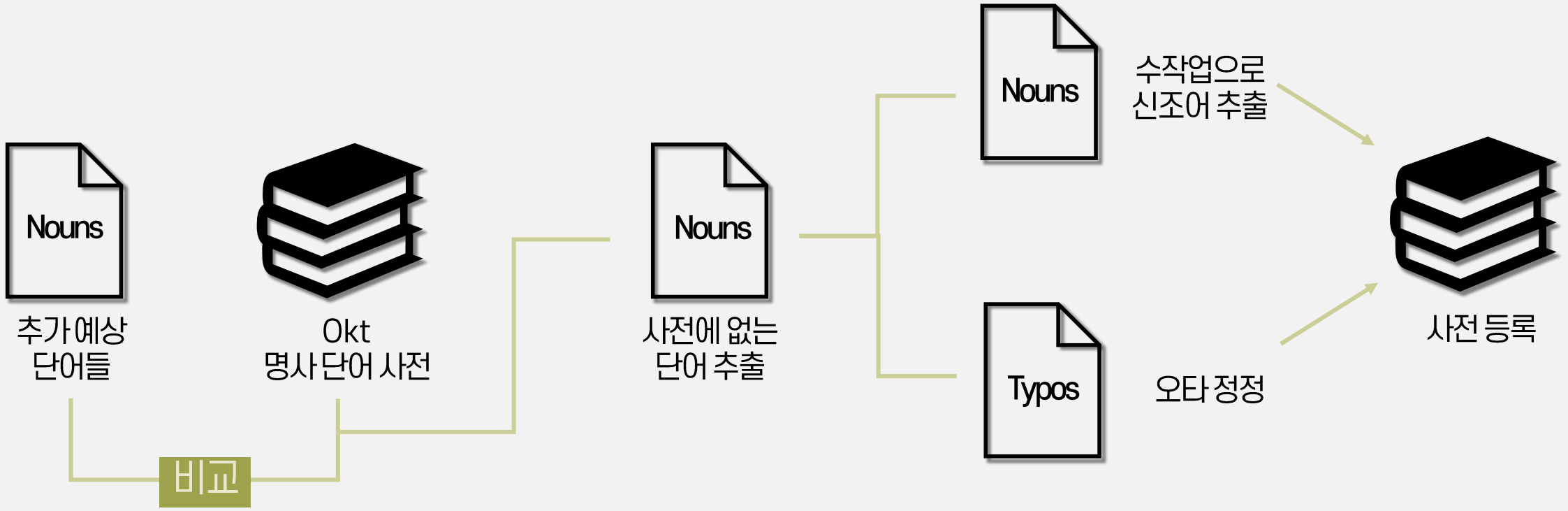
품사가
Noun인 것만
추출

Nouns

추가 예상
단어들

SoynIp를 통해 준비한 명사 데이터와 Okt의 단어사전을 비교하여 okt에 추가할 새로운 단어를 추출합니다

추가 단어 선정 과정 도식화



SoynIp를 통해 준비한 명사 데이터와 Okt의 단어사전을 비교하여 okt에 추가할 새로운 단어를 추출합니다

```
140 가볍고 1
141 가볍고 보드랍게 1
142 가볍고 상쾌하다 2
143 가볍고 상쾌한 2
144 가볍고 시원하게 2
145 가볍고 편안하게 2
146 가볍고 환하게 2
147 가분가분 1
148 가분히 1
149 가뿐가뿐 1
150 가뿐가뿐하다 1
151 가뿐가뿐히 1
152 가뿐하게 1
153 가뿐하다 1
154 가뿐한 1
155 가뿐한 느낌 1
156 가뿐한 느낌이 1
```


라벨링

인간의 기본적 감정 vs 영화 리뷰에서의 긍 /부정

영화 리뷰 긍/부정 분류 예시

리뷰 예시	KNU 감성사전	사용자 지정
완전 슬픔.. 오열함ㅜㅜㅜㅜ	부정으로 분류	긍 / 부정 없음
개존잼 진심 마블리 완전 죠앙	개존잼, 죠앙 단어 없음	긍정으로 분류

▶ 영화에서는 다양한 이야기가 존재 (ex. 범죄, 살인, 눈물 등)

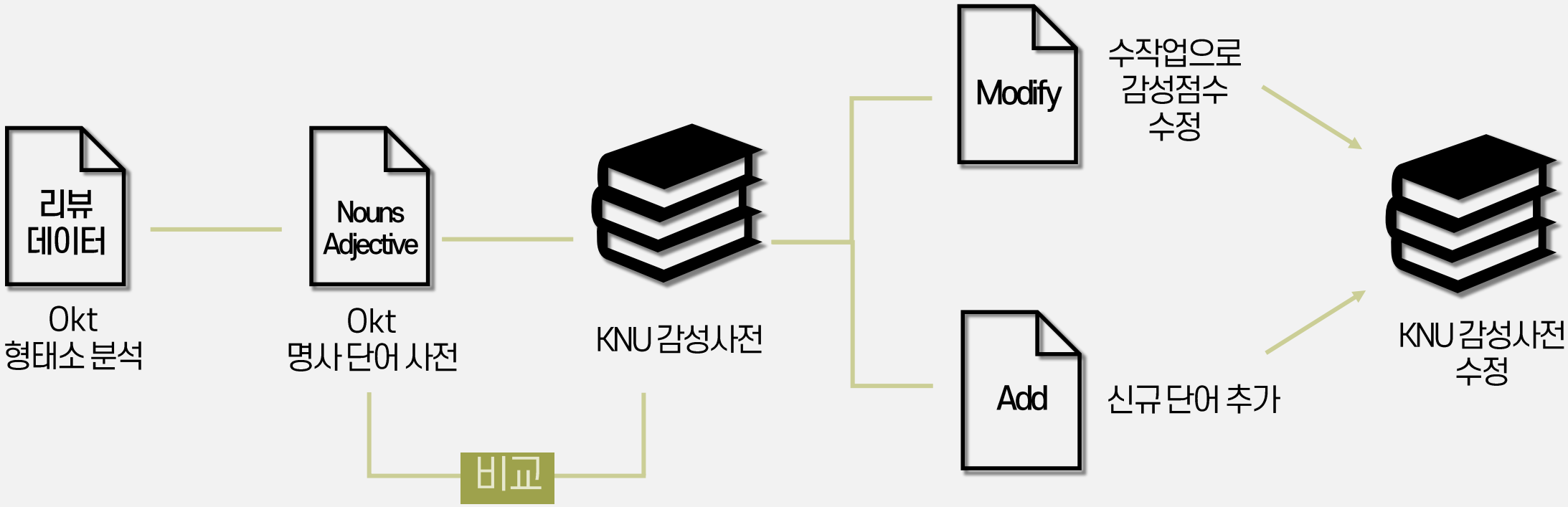
→ 만약 **줄거리와 관련해서 자신의 감정에 대해 서술했다면** 영화의 평과 **혼동**됨

▶ 같은 의미이더라도 다양한 언어 존재 (ex. 잼따, 잼씀, 재미따, 잼썼어, 재밌었다)

→ 리뷰는 **특히 줄임말, 신조어, 오타 등 다양하게 존재**. 이런 상황도 점수에 반영되어야 함.

리뷰 데이터를 Okt로 형태소 분석한 후 KNU 감성사전과 비교하여 일치 / 불일치 데이터를 생성합니다

추가 단어 선정 과정 도식화



라벨링_감성사전 수정하기

일치 / 불일치 데이터의 긍 / 부정을 수정하여 리뷰 데이터 분석에 적합한 감성사전을 준비합니다

감성사전과 일치하는 데이터 수정

낡다	-1	0
구토	-1	0
함부로	-1	0
버럭	-1	0
칙칙하다	-1	-1
아찔하다	-1	-1
흑흑	-1	1
섭섭하다	-1	-1
고함	-1	0
고리타분하다	-1	-1
무모하다	-1	-1
오해	-1	0

줄거리와 관련 있는 단어의
긍/부정 수정을 목적으로
사용자 변경이 필요한 것들을 수정

감성사전과 불일치하는 데이터 수정

별루	-1
별루더	-1
별루였	-1
별루임	-1
별미	1
병맛	-1
병쉰	-1
보세욬	1
보셈	1
보셈미쳣음	1

긍정이면 1, 부정이면 -1로
감성사전에 등록되지 않은
단어에 대한 긍/부정을 입력

KNU 감성사전

감성점수 계산 예시(긍정 / 부정)

(Example 1) - 민식이 형님 개멋져요 사랑합니다
> 민 / 식이 / 형님 / 개 / 멋지다 / 사랑 / 하다

(감성사전 매칭 결과)
> 사랑하다(2), 멋지다(2) >>> 감성점수 **4점**

(Example 2) - 최악의 영화 돈아까 움 시간 아까움내 정신에도 안조음진짜 최악
> 최악 / 의 / 영화 / 돈 / 아깝다 / 움 / 시간 / 아깝다 / 움 / 내 / 정신 / 에도 / 안좋다 / 진짜 / 최악

(감성사전 매칭 결과)
> 최악의(-2), 아깝다(-1)*2, 최악(-2) >>> 감성점수 **-6점**

라벨링_긍부정으로 라벨링

전체 리뷰의 감성점수로 전체 리뷰의 긍 / 부정, 중립을 라벨링하고, 평점 또한 중간점수를 기점으로 긍/부정, 중립을 라벨링합니다

감성점수로 긍 / 부정 라벨링

중립

부정 < 0 < 긍정

평점을 기준으로 긍 / 부정 라벨링

중립

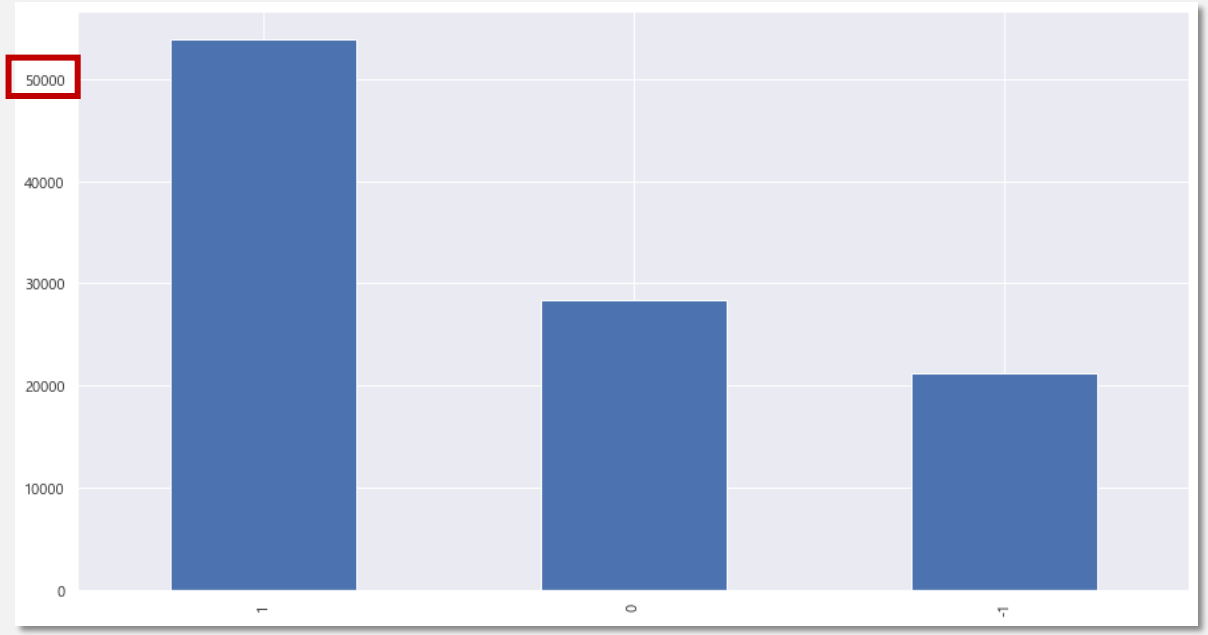
부정 < 5 < 긍정

시각화 & 인사이트 도출

시각화 & 인사이트 도출

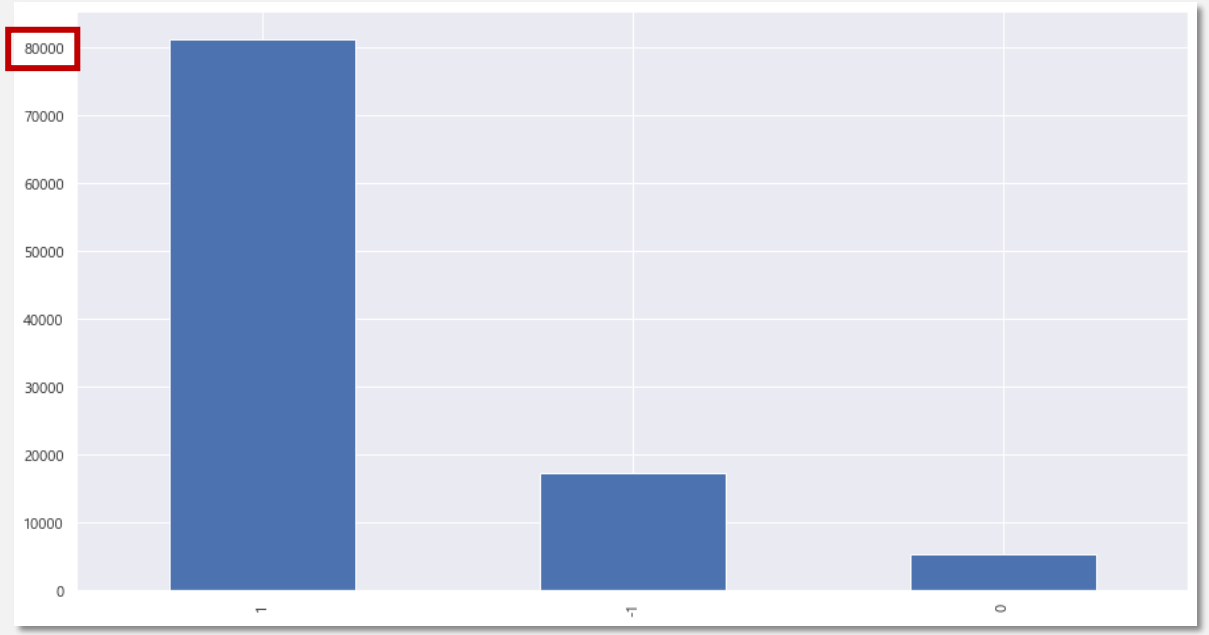
리뷰의 감성점수와 평점의 긍 / 부정을 비교하였을 때,
평점은 감성점수에 비해 부정이 적고, 긍정이 많음

감성점수 긍/부정, 중립 분포



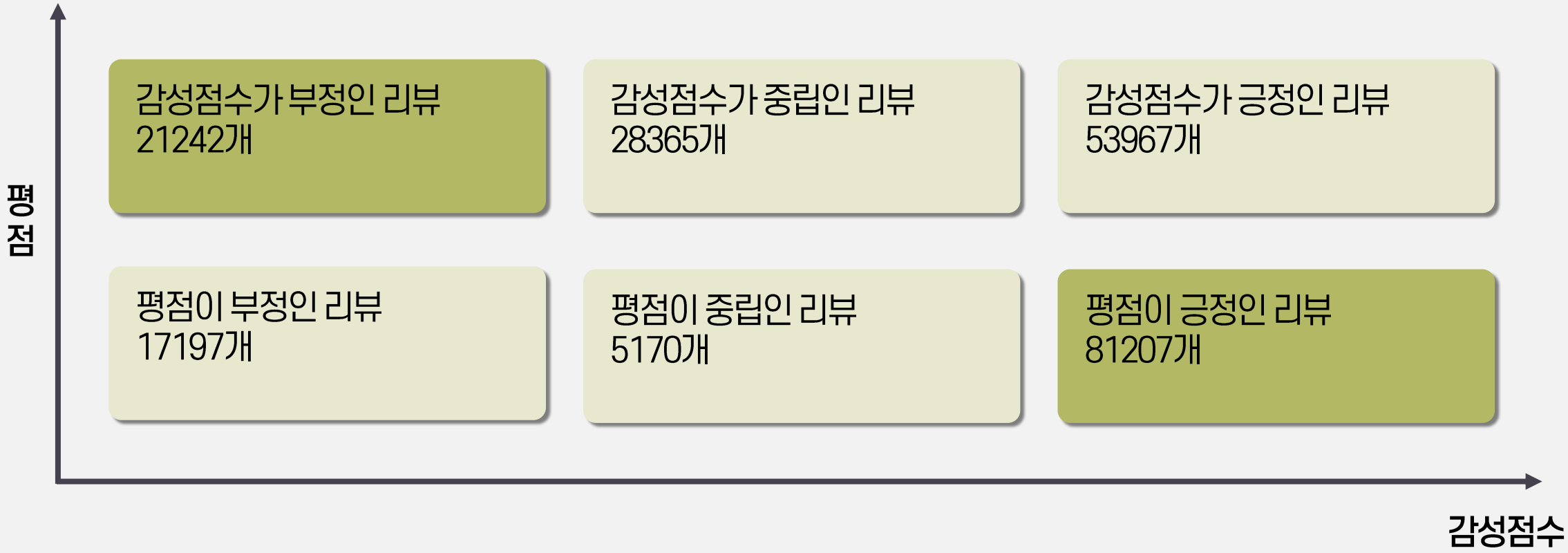
긍정 중립 부정

평점 긍/부정, 중립 분포



긍정 부정 중립

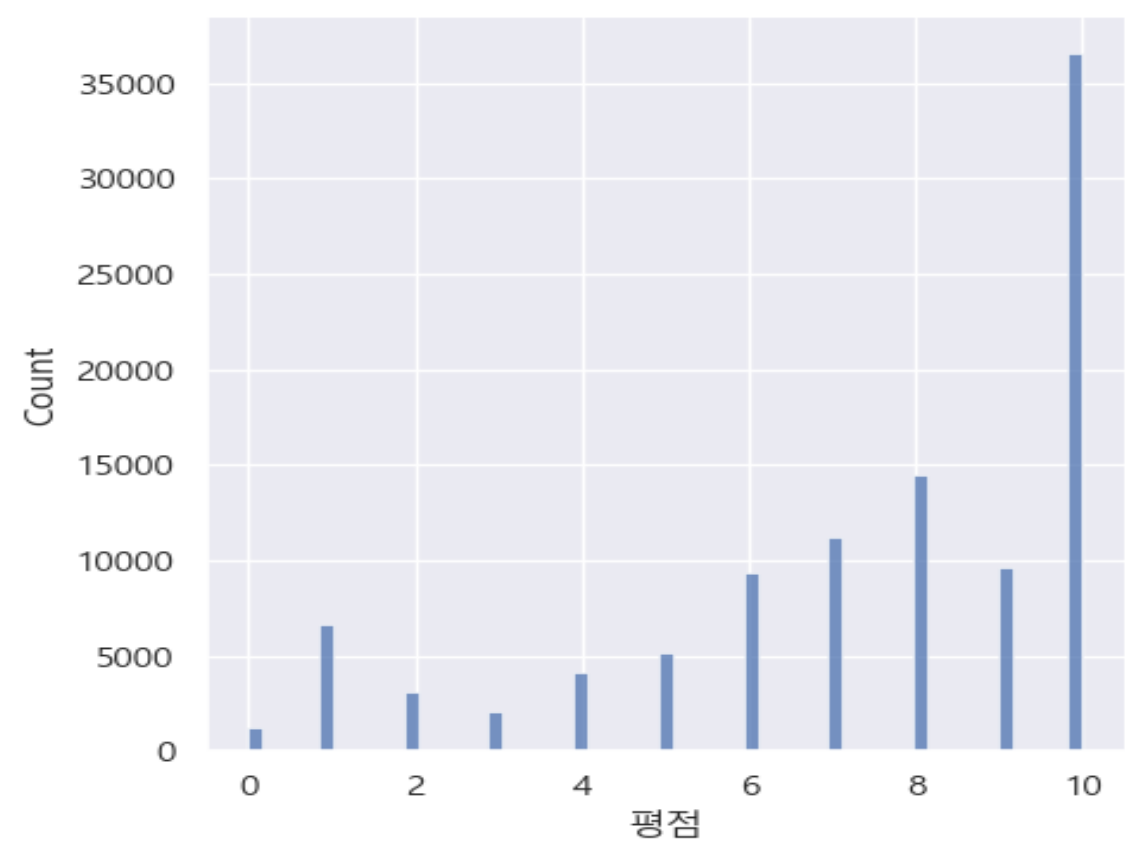
시각화 & 인사이트 도출



실제로 느낀 것보다 **평점을 후하게 준다**는 것을 알 수 있음

시각화 & 인사이트 도출

평점별 리뷰수 히스토그램



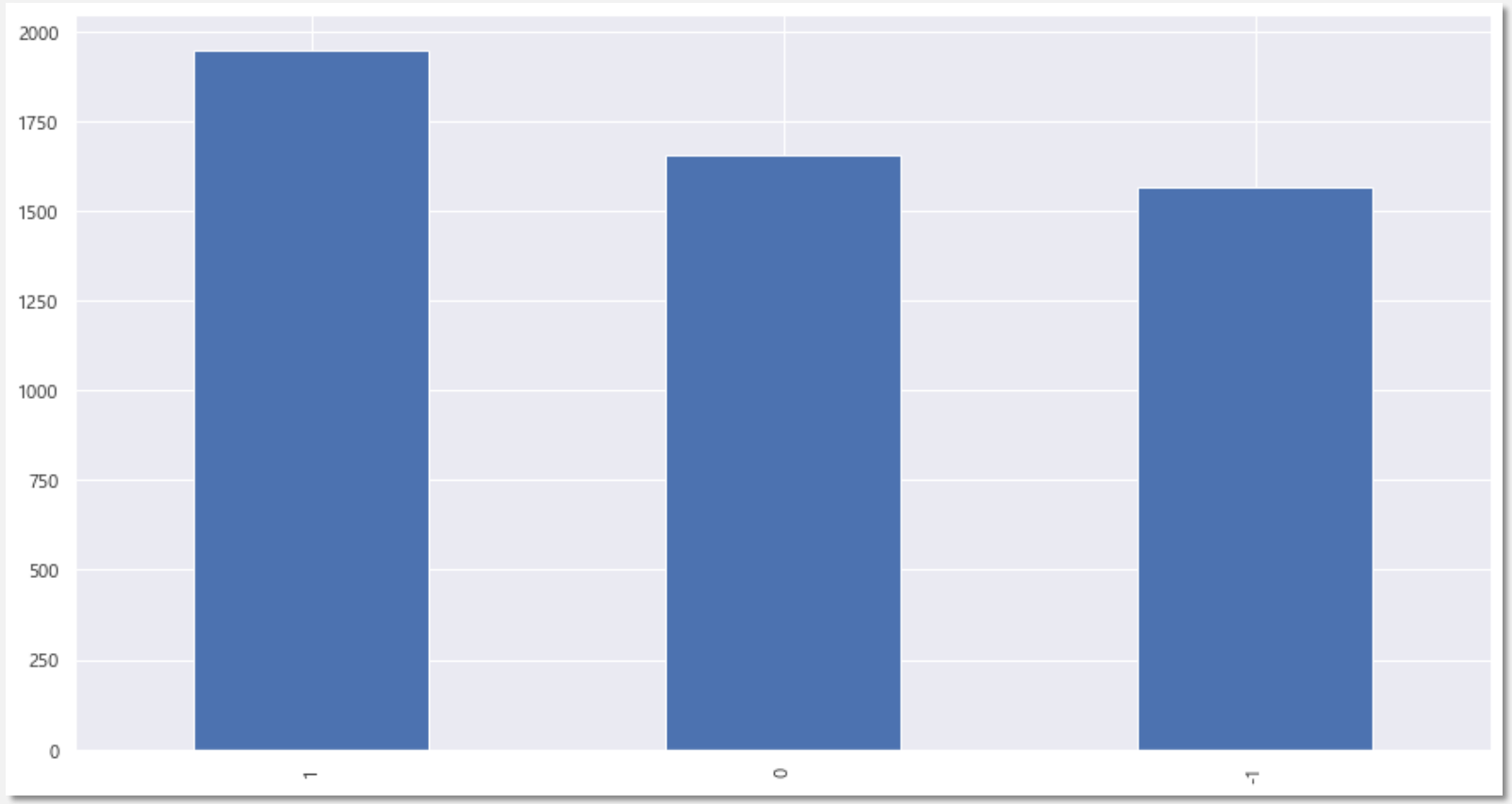
높은 평점을 남길수록 리뷰를 남기는 확률 ↑
평점이 전체적으로 높게 나올 가능성

더 정확한 판단을 위해
리뷰 텍스트 분석을 통한 감성분석 필요

시각화 & 인사이트 도출

평점을 중간점(5점)으로 준 사람들의 감성점수 긍 / 부정 분포를 보니 긍정적으로 평가하는 경우가 많음

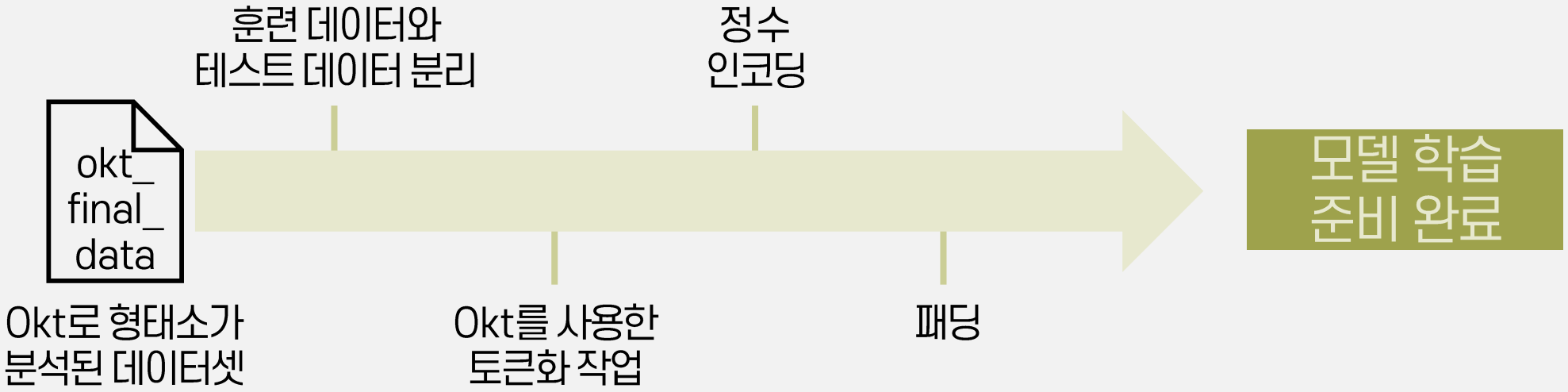
긍/부정, 중립별 리뷰 수



긍정 38%
중립 32%
부정 30%

모델

모델 학습을 위해 필요한 작업을 진행합니다



모델

앞선 단계에서 생성한 시퀀스 데이터를 학습하기 위해 양방향 LSTM 계층을 정의 후 모델을 생성합니다

```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_9 (Embedding)     (None, 60, 128)           3206528

 dropout_5 (Dropout)         (None, 60, 128)           0

 lstm_9 (LSTM)               (None, 128)               131584

 dense_11 (Dense)            (None, 1)                 129

=================================================================
Total params: 3,338,241
Trainable params: 3,338,241
Non-trainable params: 0
_________________________________________________________________
```

Vocab_size = 25051

Embedding_dim = 128

hidden_units = 128

Input_length = max_len

Dropout(0.5)

Bidirectional 사용

Return_sequences = True

TimeDistributed

Bi-LSTM으로 학습한 결과, 정확도는 71.72%, loss는 0.5963

다양한 모델 학습을 통한 비교를 위해 LSTM 계층을 정의 후 모델을 생성합니다

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 60, 128)           3206528

 dropout_1 (Dropout)         (None, 60, 128)           0

 lstm_1 (LSTM)               (None, 128)               131584

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 3,338,241
Trainable params: 3,338,241
Non-trainable params: 0
_________________________________________________________________
```

Vocab_size = 25051

Embedding_dim = 128

hidden_units = 128

Input_length = max_len

Dropout(0.5)

EarlyStopping

LSTM으로 학습한 결과, 정확도는 91.64%, loss는 0.2212

다양한 모델 학습을 통한 비교를 위해 CNN 계층을 정의 후 모델을 생성합니다

```
Model: "model"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_1 (InputLayer)         [(None, 60)]              0           []

embedding (Embedding)        (None, 60, 128)           3206528     ['input_1[0][0]']

dropout (Dropout)            (None, 60, 128)           0           ['embedding[0][0]']

conv1d (Conv1D)              (None, 58, 128)           49280       ['dropout[0][0]']

conv1d_1 (Conv1D)            (None, 57, 128)           65664       ['dropout[0][0]']

conv1d_2 (Conv1D)            (None, 56, 128)           82048       ['dropout[0][0]']

global_max_pooling1d (GlobalMa  (None, 128)            0           ['conv1d[0][0]']
xPooling1D)

global_max_pooling1d_1 (Global  (None, 128)            0           ['conv1d_1[0][0]']
MaxPooling1D)

global_max_pooling1d_2 (Global  (None, 128)            0           ['conv1d_2[0][0]']
MaxPooling1D)

concatenate (Concatenate)    (None, 384)               0           ['global_max_pooling1d[0][0]',
                                                                   'global_max_pooling1d_1[0][0]',
                                                                   'global_max_pooling1d_2[0][0]']

dropout_1 (Dropout)          (None, 384)               0           ['concatenate[0][0]']

dense_6 (Dense)              (None, 128)               49280       ['dropout_1[0][0]']

dense_7 (Dense)              (None, 1)                 129         ['dense_6[0][0]']

=================================================================================================
Total params: 3,452,929
Trainable params: 3,452,929
Non-trainable params: 0
_________________________________________________________________________________________________
```

- Vocab_size = 25051
- Embedding_dim = 128
- hidden_units = 128
- Input_length = max_len
- Dropout(0.5, 0.8)
- MaxPooling1D
- Concatenate

CNN으로 학습한 결과, 정확도는 91.86%, loss는 0.2023

모델

가장 accuracy가 높은 CNN 모델을 사용하여 새로운 리뷰에 대한 감성분석을 예측하여 봅니다

```
1 sentiment_predict('세시간동안 전혀 지루함 못 느낌. 서사, 볼거리, 메시지 등 다 최고입니다')
```

```
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 15ms/step
98.07% 확률로 긍정 리뷰입니다.
```

```
1 sentiment_predict('기대가 컷던탓인지 눈은즐거웠으나스토리가 아쉬움')
```

```
1/1 [==============================] - 0s 25ms/step
1/1 [==============================] - 0s 16ms/step
92.67% 확률로 부정 리뷰입니다.
```

```
1 sentiment_predict('연기력이 살린 영화 담아내려는 스토리가 너무 많았던것같아요')
```

```
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 16ms/step
95.77% 확률로 긍정 리뷰입니다.
```

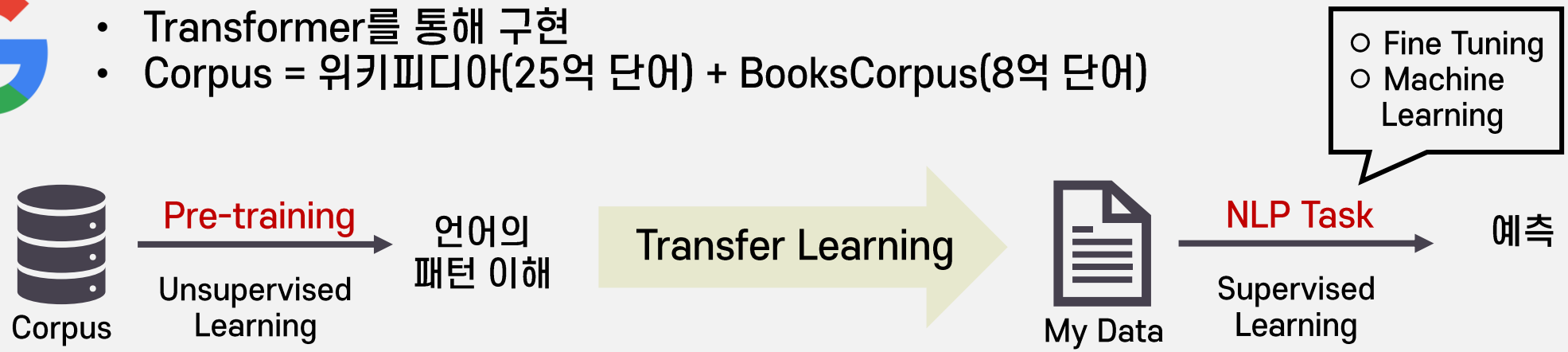
```
1 sentiment_predict('개별로임 영화 개노잼 그냥 돈아까움 볼 가치가 없는 영화임')
```

```
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 16ms/step
99.97% 확률로 부정 리뷰입니다.
```

새로운 리뷰에 대해서도 감성분석이 어느정도 잘 이뤄지는 것을 확인 가능

B E R T
(Bidirectional Encoder Representations from Transformer)

- Transformer를 통해 구현
- Corpus = 위키피디아(25억 단어) + BooksCorpus(8억 단어)



▶ 정보 손실 방지, ' 강조'의 개념 **문맥**을 이해하기 위함

Attention Mechanism → **Transformer**

▶ Transformer의 인코더를 발전시킴

BERT → **KoBERT** **KcBERT** **...**

B E R T
(Bidirectional Encoder Representations from Transformer)

BERT 실행 순서

1. Tokenizer에 Pre-training 불러오기
2. 각 리뷰에 처음과 끝에 [CLS], [SEP] 토큰 추가
3. Tokenizing
4. Padding
5. Attention Mask부여
   (패딩을 인식하게 해서 필요없는 연산을 줄임)
6. 학습 및 평가
7. 새로운 데이터로 예측

예측

Sent1 = '세시간동안 전혀 지루함 못 느낌. 서사, 볼거리, 메시지 등 다 최고입니다'
Sent2 = '기대가 컷던탓인지 눈은즐거웠으나스토리가 아쉬움'
Sent3 = '연기력이 살린 영화 담아내려는 스토리가 너무 많았던것같아요'
Sent4 = '개별로임 영화 개노잼 그냥 돈아까움 볼 가치가 없는 영화임'

	문장1	문장2	문장3	문장4
BERT	긍(98.27)	부(97.67)	긍(98.78)	부(97.41)
KoBERT	긍(99.02)	긍(99.01)	긍(99.03)	부(98.21)

BERT 모형 분석 요약

Model	Accuracy
CNN	0.919
LSTM	0.916
BiLSTM	0.717
BERT	0.995
KoBERT	0.980

1. Accuracy로 본 성능
   BERT > KoBERT > CNN > LSTM > BiLSTM
2. 기존의 딥러닝보다 Transformer를 이용한 BERT 모형의 성능이 우수함을 확인
3. 새로운 리뷰에 대해 긍 / 부정으로 잘 파악하는 것 같으나, 긍 / 부정이 섞인 리뷰의 경우는 애매모호함.
4. BERT의 성능이 더 좋은 것으로 나왔지만, 시간 비용적인면에서 KoBERT가 더 우수하다.

결론 & 제언

요약

- **특정 도메인 감성 사전의 필요성**
  - 라벨링 되어있지 않은 데이터에 대해 감성사전을 기반으로 분류해본 결과,
    생각보다 **많은 부분에서 이슈**가 있었다. (띄어쓰기, 맞춤법, 신조어, 도메인 특성 등)
  - 수작업으로 분류했던 부분이 2곳. 오류가 있을 가능성 높다. → 교차 검증 필요

- **라벨과 별점의 차이**
  - 평점이 중간점수인 5점으로 주었을 때는 긍정을 나타낼 확률이 높다
  - 실제로 느낀 것보다 평점을 후하게 주는 경향이 있다
  - 높은 평점을 남길수록 리뷰를 남기는 확률이 높기에 평점이 전체적으로 높게 나올 가능성이 있다

- **앞으로의 과제: 특정 도메인에 대한 학습 강화 / BERT를 이용한 다중 분류로 확장**
  - BERT의 등장으로 기존의 신조어에 대한 문제와 문맥의 이해 문제 해결 가능성이 보인다.
  - 리뷰 뿐만 아니라 특정 도메인에 맞춰 많은 연구가 계속되야 할 것으로 보인다.
  - 감정을 더 세분화한다면 상황에 맞는 보다 효율적인 의사결정이 가능할 것으로 보인다.

결론 & 제언

리뷰 감성분석 결과를 영화 선호도 평가 기준으로 사용한다면 여러가지 서비스에 적용될 수 있습니다.

영화 추천 서비스

개별 영화에 대한 긍/부정 정도를 파악하여
선호하는 장르와 동일한 영화 중
선호도가 높은 영화 추천 가능

OTT 서비스 영화 계약 기준

YouTube TVING NETFLIX WATCHA wavve Disney+

리뷰 감성분석한 결과
높은 선호도를 보이는 영화를 선정하여
계약을 맺거나 연장할 수 있는 ott 서비스

Reference

- 장연지, 최지선 and 김한샘. (2022). 감정 어휘 사전을 활용한 KcBert 기반 영화 리뷰 말뭉치 감정 분석. 정보과학회논문지, 49(8), 608-616.
- 조정태 and 최상현. (2015). 영화리뷰 감성 분석을 통한 평점 예측 연구. 경영과 정보연구, 34(3), 161-177.
- 김지현, 하희정, 김서희 and 정영욱. (2021). OTT 서비스 콘텐츠 추천 사용자 경험 분석 -넷플릭스 사례를 중심으로. Journal of Integrated Design Research, 20(2), 73-87.
- Soynlp, https://github.com/lovit/soynlp/blob/master/tutorials/nounextractor-v2_usage.ipynb
- KNU 한국어 감성사전, https://github.com/lovit/soynlp , https://github.com/park1200656/KnuSentiLex
- 딥러닝을 이용한 자연어 입문, https://wikidocs.net/92961
- BERT, https://github.com/deepseasw/bert-naver-movie-review/blob/master/bert_naver_movie.ipynb
- KoBERT, https://github.com/SKTBrain/KoBERT/blob/master/scripts/NSMC/naver_review_classifications_pytorch_kobert.ipynb